

# Примена машинског учења у статичкој верификацији софтвера

Семинарски рад у оквиру курса  
Методологија стручног и научног рада  
Математички факултет

Лазар Ранковић, Немања Мићовић, Урош Стегић  
lazar.rankovic@outlook.com, nmicovic@outlook.com, mi10287@alas.matf.bg.ac.rs

## Абстракт

Пошто ћемо абстракт писати на крају, онда док радимо да искористимо ово за интерне потребе. До сада прегледани радови:

- Finding latent code errors via machine learning over program executions [1]
- Learning Invariants using Decision Trees [6]
- ICE: A Robust Framework for Learning Invariants [3]
- Interpolants as Classifiers [8]
- Advanced Verification Techniques Based on learning [4]
- A Survey of Static Program Analysis Techniques [9]
- A Survey of Automated Techniques for Formal Software Verification [2]
- Regresiona verifikacija softvera korišćenjem sistema LAV [5]

## Садржај

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Верификација софтвера</b>	<b>2</b>
<b>3</b>	<b>Технике статичке верификације</b>	<b>2</b>
<b>4</b>	<b>Машинско учење</b>	<b>2</b>
4.1	Основе машинског учења . . . . .	3
<b>5</b>	<b>Веза верификације и МЛ</b>	<b>3</b>
<b>6</b>	<b>Неке примене техника машинског учења у статичкој верификацији</b>	<b>3</b>
<b>7</b>	<b>Закључак</b>	<b>3</b>
	<b>Literatura</b>	<b>3</b>

## 1 Uvod

Увод ћемо пред крај писати.

## 2 Верификација софтвера

Питања се извлаче из Милениног доктората или пронаћи неки рад чисто о верификацији.

Одговара се на питања:

- Шта је верификација
- Зашто је важна
- Уопштено како се ради

По пар реченица за ове пасусе:

**Динамичка верификација** Ово не радимо у раду, само треба кратак опис. Није довољно прецизна (реф.)

**Статичка верификација** Овиме се бавимо и поента је што је она прецизна

## 3 Технике статичке верификације

Општа прича, које су врсте (набројати макар три :P) и за сваки тип по један параграф. Постоје:

- Апстрактна интерпретација
- Симболичко израчунавање
- Проверавање ограничених модела (енг. Bounded model checking)

Литература: A Survey of Static Program Analysis Techniques [9]

A Survey of Automated Techniques for Formal Software Verification [2]

Миленин докторат (ово је још и најбоље)

## 4 Машинско учење

У претходним поглављима смо представили увод у статичку верификацију софтвера. Показана је важност те области и изложене су технике верификације. Овим поглављем ћемо представити област машинског учења. Описаћемо главне аспекте ове дисциплине, покажемо њену битност и даћемо преглед важних концепата о којима ће бити више речи у поглављу 6. Овим поглављем ћемо дати одговоре на следећа питања:

- Шта је машинско учење?
- У чему је значај ове области?
- Основни принципи.

## 4.1 Основе машинског учења

**Дефиниција 1.** *“За програм кажемо да учи из искуства  $E$  кроз обављање задатка  $T$  са мером квалитета  $P$ , ако повећањем искуства  $E$  расте мера  $P$  за обављен задатак  $T$ .”*

— Tom M. Mitchell [7]

Машинско учење можемо посматрати као област рачунарства која се бави анализом алгоритама који генерализују. Са практичног аспекта, генерализација представља уопштавање закона над датим подацима. Машинско учење се дели на три подобласти: *надгледано учење*, *не-надгледано учење* и *учење условљавањем*. Подаци из којих алгоритми машинског учења уче, могу бити обележени, необележени и могу се генерисати у фази учења. Оваква природа података је основ за разликовање три наведене подобласти.[7].

Међу многим проблемима над којима су често примењивани алгоритми машинског учења, издвојићемо проблем регресије и проблем класификације. Ови проблеми су релевантни за теме којих ћемо се дотаћи у овом раду. Под проблемом класификације подразумевамо испитивање инстанце датог објекта и одређивање класе којој он припада на основу његових својстава (атрибута). Типичан пример класификације је одређивање порекла тумора (испитивање да ли је тумор малигни или бенигни) на основу његове величине. Проблем регресије представља предикцију понашања неког параметра популације за дати објекат на основу осталих атрибута тог објекта. Као пример можемо узети предикцију цене стамбеног објекта на основу његове величине, броја соба и разних других релевантних карактеристика.

## 5 Веза верификације и МЛ

Шта су проблеми верификације, због чега се кочи, који су изазови. Неки начини решавања (паралелизација, хеуристике...) Описати који су делови верификације где лепо легне МЛ (увод у следеће поглавље).

## 6 Неке примене техника машинског учења у статичкој верификацији

Ово је есенција. Одабирају се проблеми из претходног поглавља и показује се како се решава. Прво иде неки уводни део, онда из литературе се покупе те технике и таксативно се наводе (принцип проблем-решење).

## 7 Закључак

Овде машти на вољу.. :)

## Литература

- [1] Yuriy Brun. Finding latent code errors via machine learning over program executions, 2004.

- [2] Vijay D'Silva, Daniel Kroening, and Georg Weissenbacher. A survey of automated techniques for formal software verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 27(7):1165–1178, July 2008.
- [3] Pranav Garg, Christof Löding, P. Madhusudan, Daniel Neider, and Roderick Bloem. *"ICE: A Robust Framework for Learning Invariants"*, pages "69–87". "Springer International Publishing", "Cham", "2014".
- [4] Jawahar Jain, Rajarshi Mukherjee, and Masahiro Fujita. Advanced verification techniques based on learning. In *Proceedings of the 32Nd Annual ACM/IEEE Design Automation Conference, DAC '95*, pages 420–426, New York, NY, USA, 1995. ACM.
- [5] Milena Vujosevic Janicic. Regresiona verifikacija softvera korišćenjem sistema lav.
- [6] Siddharth Krishna, Christian Puhersch, and Thomas Wies. Learning invariants using decision trees. *CoRR*, abs/1501.04725, 2015.
- [7] Tom M. Mitchell. *Machine Learning*, volume 1. McGraw Hill, 1997.
- [8] Rahul Sharma, Aditya V. Nori, and Alex Aiken. Interpolants as classifiers.
- [9] Wolfgang Wögerer and Technische Universität Wien. A survey of static program analysis techniques, 2005.